Knowledge Base Completion via Coupled Path Ranking

Quan Wang[†], **Jing Liu**[‡], **Yuanfei Luo**[†], **Bin Wang**[†], **Chin-Yew Lin**[‡]
[†]Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
{wangquan,luoyuanfei,wangbin}@iie.ac.cn
[‡]Microsoft Research, Beijing 100080, China
{liudani,cyl}@microsoft.com

Abstract

Knowledge bases (KBs) are often greatly incomplete, necessitating a demand for KB completion. The path ranking algorithm (PRA) is one of the most promising approaches to this task. Previous work on PRA usually follows a single-task learning paradigm, building a prediction model for each relation independently with its own training data. It ignores meaningful associations among certain relations, and might not get enough training data for less frequent relations. This paper proposes a novel multi-task learning framework for PRA, referred to as coupled PRA (CPRA). It first devises an agglomerative clustering strategy to automatically discover relations that are highly correlated to each other, and then employs a multi-task learning strategy to effectively couple the prediction of such relations. As such, CPRA takes into account relation association and enables implicit data sharing among them. We empirically evaluate CPRA on benchmark data created from Freebase. Experimental results show that CPRA can effectively identify coherent clusters in which relations are highly correlated. By further coupling such relations, CPRA significantly outperforms PRA, in terms of both predictive accuracy and model interpretability.

1 Introduction

Knowledge bases (KBs) like Freebase (Bollacker et al., 2008), DBpedia (Lehmann et al., 2014), and NELL (Carlson et al., 2010) are extremely useful resources for many NLP tasks (Cucerzan, 2007; Schuhmacher and Ponzetto, 2014). They provide large collections of facts about entities and their relations, typically stored as (*head entity*, *relation*, *tail entity*) triples, *e.g.*, (Paris, capitalOf, France). Although such KBs can be impressively large, they are still quite incomplete and missing crucial facts, which may reduce their usefulness in downstream tasks (West et al., 2014; Choi et al., 2015). KB completion, *i.e.*, automatically inferring missing facts by examining existing ones, has thus attracted increasing attention. Approaches to this task roughly fall into three categories: (i) path ranking algorithms (PRA) (Lao et al., 2011); (ii) embedding techniques (Bordes et al., 2013; Guo et al., 2015); and (iii) graphical models such as Markov logic networks (MLN) (Richardson and Domingos, 2006). This paper focuses on PRA, which is easily interpretable (as opposed to embedding techniques) and requires no external logic rules (as opposed to MLN).

The key idea of PRA is to explicitly use paths connecting two entities to predict potential relations between them. In PRA, a KB is encoded as a graph which consists of a set of heterogeneous edges. Each edge is labeled with a relation type that exists between two entities. Given a specific relation, random walks are first employed to find paths between two entities that have the given relation. Here a path is a sequence of relations linking two entities, *e.g.*, $h \xrightarrow{\texttt{bornIn}} e \xrightarrow{\texttt{capitalOf}} t$. These paths are then used as features in a binary classifier to predict if new instances (*i.e.*, entity pairs) have the given relation.

While KBs are naturally composed of multiple relations, PRA models these relations separately during the inference phase, by learning an individual classifier for each relation. We argue, however, that it will be beneficial for PRA to model certain relations in a collective way, particularly when the relations are closely related to each other. For example, given two relations bornIn and livedIn,

Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 1308–1318,
Berlin, Germany, August 7-12, 2016. ©2016 Association for Computational Linguistics

there must be a lot of paths (features) that are predictive for both relations, *e.g.*, $h \xrightarrow{\texttt{nationality}} e \xrightarrow{\texttt{hasCapital}} t$. These features make the corresponding relation classification tasks highly related. Numerous studies have shown that learning multiple related tasks simultaneously (a.k.a. multi-task learning) usually leads to better predictive performance, profiting from the relevant information available in different tasks (Carlson et al., 2010; Chapelle et al., 2010).

This paper proposes a novel multi-task learning framework that couples the path ranking of multiple relations, referred to as coupled PRA (CPRA). The new model needs to answer two critical questions: (i) *which relations should be coupled*, and (ii) *in what manner they should be coupled*.

As to the first question, it is obvious that not all relations are suitable to be learned together. For instance, modeling `bornIn` together with `hasWife` might not bring any real benefits, since there are few common paths between these two relations. CPRA introduces a common-path based similarity measure, and accordingly devises an agglomerative clustering strategy to group relations. Only relations that are grouped into the same cluster will be coupled afterwards.

As to the second question, CPRA follows the common practice of multi-task learning (Evgeniou and Pontil, 2004), and couples relations by using classifiers with partially shared parameters. Given a cluster of relations, CPRA builds the classifiers upon (i) relation-specific parameters to address the specifics of individual relations, and (ii) shared parameters to model the commonalities among different relations. These two types of parameters are balanced by a coupling coefficient, and learned jointly for all relations. In this way CPRA couples the classification tasks of multiple relations, and enables implicit data sharing and regularization.

The major contributions of this paper are as follows. (i) We design a novel framework for multi-task learning with PRA, *i.e.*, CPRA. To the best of our knowledge, this is the first study on multi-task PRA. (ii) We empirically verify the effectiveness of CPRA on a real-world, large-scale KB. Specifically, we evaluate CPRA on benchmark data created from Freebase. Experimental results show that CPRA can effectively identify coherent clusters in which relations are highly correlated. By further coupling such relations, CPRA substantially outperforms PRA, in terms of not only predictive accuracy but also model interpretability. (iii) We compare CPRA and PRA to the embedding-based TransE model (Bordes et al., 2013), and demonstrate their superiority over TransE. As far as we know, this is the first work that formally compares PRA-style approaches to embedding-based ones, on publicly available Freebase data.

In the remainder of this paper, we first review related work in Section 2, and formally introduce PRA in Section 3. We then detail the proposed CPRA framework in Section 4. Experiments and results are reported in Section 5, followed by the conclusion and future work in Section 6.

2 Related Work

We first review three lines of related work: (i) KB completion, (ii) PRA and its extensions, and (iii) multi-task learning, and then discuss the connection between CPRA and previous approaches.

KB completion. This task is to automatically infer missing facts from existing ones. Prior work roughly falls into three categories: (i) path ranking algorithms (PRA) which use paths that connect two entities to predict potential relations between them (Lao et al., 2011; Lao and Cohen, 2010); (ii) embedding-based models which embed entities and relations into a latent vector space and make inferences in that space (Nickel et al., 2011; Bordes et al., 2013); (iii) probabilistic graphical models such as the Markov logic network (MLN) and its variants (Pujara et al., 2013; Jiang et al., 2012). This paper focuses on PRA, since it is easily interpretable (as opposed to embedding-based models) and requires no external logic rules (as opposed to MLN and its variants).

PRA and its extensions. PRA is a random walk inference technique designed for predicting new relation instances in KBs, first proposed by Lao and Cohen (2010). Recently various extensions have been explored, ranging from incorporating a text corpus as additional evidence during inference (Gardner et al., 2013; Gardner et al., 2014), to introducing better schemes to generate more predictive paths (Gardner and Mitchell, 2015; Shi and Weninger, 2015), or using PRA in a broader context such as Google's Knowledge Vault (Dong et al., 2014). All these approaches are based on some single-task version of PRA, while our work explores multi-task learning for it.

Multi-task learning. Numerous studies have shown that learning multiple related tasks simultaneously can provide significant benefits relative to learning them independently (Caruana, 1997). A key ingredient of multi-task learning is to model the notion of task relatedness, through either parameter sharing (Evgeniou and Pontil, 2004; Ando and Zhang, 2005) or feature sharing (Argyriou et al., 2007; He et al., 2014). In recent years, there has been increasing work showing the benefits of multi-task learning in NLP-related tasks, such as relation extraction (Jiang, 2009; Carlson et al., 2010) and machine translation (Sennrich et al., 2013; Cui et al., 2013; Dong et al., 2015). This paper investigates the possibility of multi-task learning with PRA, in a parameter sharing manner.

Connection with previous methods. Actually, modeling multiple relations collectively is a common practice in embedding-based approaches. In such a method, embeddings are learned jointly for all relations, over a set of shared latent features (entity embeddings), and hence can capture meaningful associations among different relations. As shown by (Toutanova and Chen, 2015), observed features such as PRA paths usually perform better than latent features for KB completion. In this context, CPRA is designed in a way that gets the multi-relational benefit of embedding techniques while keeping PRA-style path features. Nickel et al. (2014) and Neelakantan et al. (2015) have tried similar ideas. However, their work focuses on improving embedding techniques with observed features, while our approach aims at improving PRA with multi-task learning.

3 Path Ranking Algorithm

PRA was first proposed by Lao and Cohen (2010), and later slightly modified in various ways (Gardner et al., 2014; Gardner and Mitchell, 2015). The key idea of PRA is to explicitly use paths that connect two entities as features to predict potential relations between them. Here a path is a sequence of relations $\langle r_1, r_2, \cdots, r_\ell \rangle$ that link two entities. For example, $\langle \texttt{bornIn}, \texttt{capitalOf} \rangle$ is a path linking `SophieMarceau` to `France`, through an intermediate node `Paris`. Such paths are then used as features to predict the presence of specific relations, *e.g.*, `nationality`. A typical PRA model consists of three steps: feature extraction, feature computation, and relation-specific classification.

Feature extraction. The first step is to generate and select path features that are potentially useful for predicting new relation instances. To this end, PRA first encodes a KB as a multi-relation graph. Given a pair of entities (h, t), PRA then finds the paths by performing random walks over the graph, recording those starting from h and ending at t with bounded lengths. More exhaustive strategies like breadth-first (Gardner and Mitchell, 2015) or depth-first (Shi and Weninger, 2015) search could also be used to enumerate the paths. After that a set of paths are selected as features, according to some precision-recall measure (Lao et al., 2011), or simply frequency (Gardner et al., 2014).

Feature computation. Once path features are selected, the next step is to compute their values. Given an entity pair (h, t) and a path π, PRA computes the feature value as a random walk probability $p(t|h, \pi)$, *i.e.*, the probability of arriving at t given a random walk starting from h and following exactly all relations in π. Computing these random walk probabilities could be at great expense. Gardner and Mitchell (2015) recently showed that such probabilities offer no discernible benefits. So they just used a binary value to indicate the presence or absence of each path. Similarly, Shi and Weninger (2015) used the frequency of a path as its feature value. Besides paths, other features such as path bigrams and vector space similarities could also be incorporated (Gardner et al., 2014).

Relation-specific classification. The last step of PRA is to train an individual classifier for each relation, so as to judge whether two entities should be linked by that relation. Given a relation and a set of training instances (*i.e.*, pairs of entities that are linked by the relation or not, with features selected and computed as above), one can use any kind of classifier to train a model. Most previous work simply chooses logistic regression.

4 Coupled Path Ranking Algorithm

As we can see, PRA (as well as its variants) follows a single-task learning paradigm, which builds a classifier for each relation independently with its own training data. We argue that such a single-task strategy might not be optimal for KB completion: (i) by learning the classifiers independently, it fails to discover and leverage meaningful associations among different relations; (ii) it might not perform well on less frequent relations for which only a few training instances are available. This section presents coupled PRA (CPRA), a novel multi-task learning framework that couples the path ranking of multiple relations. Through a multi-task strategy, CPRA takes into account relation association and enables implicit data sharing among them.

4.1 Problem Formulation

Suppose we are given a KB containing a collection of triples $\mathcal{O} = \{(h, r, t)\}$. Each triple is composed of two entities $h, t \in \mathcal{E}$ and their relation $r \in \mathcal{R}$, where $\mathcal{E}$ is the entity set and $\mathcal{R}$ the relation set. The KB is then encoded as a graph $\mathcal{G}$, with entities represented as nodes, and triple (h, r, t) a directed edge from node h to node t. We formally define KB completion as a binary classification problem. That is, given a particular relation r, for any entity pair (h, t) such that $(h, r, t) \notin \mathcal{O}$, we would like to judge whether h and t should be linked by r, by exploiting the graph structure of $\mathcal{G}$. Let $\overline{\mathcal{R}} \subseteq \mathcal{R}$ denote a set of relations to be predicted.

Each relation $r \in \overline{\mathcal{R}}$ is associated with a set of training instances. Here a training instance is an entity pair (h, t), with a positive label if $(h, r, t) \in \mathcal{O}$ or a negative label otherwise.[1] For each of the entity pairs, path features could be extracted and computed using techniques described in Section 3. We denote by $\mathbf{\Pi}_r$ the set of path features extracted for relation r, and define its training set as $\mathcal{T}_r = \{(\mathbf{x}_{ir}, y_{ir})\}$. Here $\mathbf{x}_{ir}$ is the feature vector for an entity pair, with each dimension corresponding to a path $\pi \in \mathbf{\Pi}_r$, and $y_{ir} = \pm 1$ is the label. Note that our primary goal is to verify the possibility of multi-task learning with PRA. It is beyond the scope of this paper to further explore better feature extraction or computation.

Given the relations and their training instances, CPRA performs KB completion using a multi-task learning strategy. It consists of two components: relation clustering and relation coupling. The former automatically discovers highly correlated relations, and the latter further couples the learning of these relations, described in detail as follows.

4.2 Relation Clustering

It is obvious that not all relations are suitable to be coupled. We propose an agglomerative clustering algorithm to automatically discover relations that are highly correlated and should be learned together. Our intuition is that relations sharing more common paths (features) are probably more similar in classification, and hence should be coupled.

Specifically, we start with $|\overline{\mathcal{R}}|$ clusters and each cluster contains a single relation $r \in \overline{\mathcal{R}}$. Here $|\cdot|$ is the cardinality of a set. Then we iteratively merge the most similar clusters, say $\mathcal{C}_m$ and $\mathcal{C}_n$, into a new cluster $\mathcal{C}$. The similarity between two clusters is defined as:

$$\mathrm{Sim}(\mathcal{C}_i, \mathcal{C}_j) = \frac{|\mathbf{\Pi}_{\mathcal{C}_i} \cap \mathbf{\Pi}_{\mathcal{C}_j}|}{\min(|\mathbf{\Pi}_{\mathcal{C}_i}|, |\mathbf{\Pi}_{\mathcal{C}_j}|)}, \qquad (1)$$

where $\mathbf{\Pi}_{\mathcal{C}_i}$ is the feature set associated with cluster $\mathcal{C}_i$ (if $\mathcal{C}_i$ contains a single relation, $\mathbf{\Pi}_{\mathcal{C}_i}$ the feature set associated with that relation). It essentially measures the overlap between two feature sets. The larger the overlap is, the higher the similarity will be. Once two clusters are merged, we update the feature set associated with the new cluster: $\mathbf{\Pi}_{\mathcal{C}} = \mathbf{\Pi}_{\mathcal{C}_m} \cup \mathbf{\Pi}_{\mathcal{C}_n}$. The algorithm stops when the highest cluster similarity is below some predefined threshold δ. This paper empirically sets $\delta = 0.5$. As such, relations sharing a substantial number of common paths are grouped into the same cluster.

4.3 Relation Coupling

After clustering, the next step of CPRA is to couple the path ranking of different relations within each cluster, *i.e.*, to learn the classification tasks for these relations simultaneously. We employ a multi-task classification algorithm similar to (Evgeniou and Pontil, 2004), and learn the classifiers jointly in a parameter sharing manner.

Consider a cluster containing K relations $\mathcal{C} = \{r_1, r_2, \cdots, r_K\}$. Recall that during the clustering phase a shared feature set has been generated for that cluster, *i.e.*, $\mathbf{\Pi}_{\mathcal{C}} = \mathbf{\Pi}_{r_1} \cup \cdots \cup \mathbf{\Pi}_{r_K}$. We first reform the training instances for the K relations using this shared feature set, so that all training data is represented in the same space.[2] We denote by $\mathcal{T}_k = \{(\mathbf{x}_{ik}, y_{ik})\}_{i=1}^{N_k}$ the reformed training data associated with the k-th relation. Then our goal is to jointly learn K classifiers $f_1, f_2, \cdots, f_K$ such that $f_k(\mathbf{x}_{ik}) \approx y_{ik}$.

We first assume that the classifier for each relation has a linear form $f_k(\mathbf{x}) = \mathbf{w}_k \cdot \mathbf{x} + b_k$, where $\mathbf{w}_k \in \mathbb{R}^d$ is the weight vector and b_k the bias. To model associations among different relations, we further assume that all $\mathbf{w}_k$ and b_k can be written, for every $k \in \{1, \cdots, K\}$, as:

$$\mathbf{w}_k = \mathbf{w}_0 + \mathbf{v}_k \;\text{ and }\; b_k = b_0. \qquad (2)$$

Here the shared $\mathbf{w}_0$ is used to model the commonalities among different relations, and the relation-specific $\mathbf{v}_k$ to address the specifics of individual

[1] We will introduce the details of generating negative training instances in Section 5.1.

[2] Note that $\mathbf{\Pi}_{r_k} \subseteq \mathbf{\Pi}_{\mathcal{C}}$. We just assign zero values to features that are contained in $\mathbf{\Pi}_{\mathcal{C}}$ but not in $\mathbf{\Pi}_{r_k}$.

relations. If the relations are closely related ($\mathbf{v}_k \approx \mathbf{0}$), they will have similar weights ($\mathbf{w}_t \approx \mathbf{w}_0$) on the common paths. We use the same bias b_0 for all the relations.[3]

We estimate $\mathbf{v}_k$, $\mathbf{w}_0$, and b_0 simultaneously in a joint optimization problem, defined as follows.

Problem 1 *CPRA amounts to solving the general optimization problem:*

$$\min_{\{\mathbf{v}_k\},\mathbf{w}_0,b_0} \sum_{k=1}^{K}\sum_{i=1}^{N_k} \ell\left(\mathbf{x}_{ik}, y_{ik}\right) + \frac{\lambda_1}{K}\sum_{k=1}^{K} \|\mathbf{v}_k\|_2^2 + \lambda_2 \|\mathbf{w}_0\|_2^2,$$

where $\ell\left(\mathbf{x}_{ik}, y_{ik}\right)$ is the loss on a training instance. It can be instantiated into a logistic regression (L-R) or support vector machine (SVM) version, by respectively defining the loss $\ell\left(\mathbf{x}_{ik}, y_{ik}\right)$ as:

$$\ell\left(\mathbf{x}_{ik}, y_{ik}\right) = \log\left(1 + \exp\left(-y_{ik} f_k(\mathbf{x}_{ik})\right)\right),$$
$$\ell\left(\mathbf{x}_{ik}, y_{ik}\right) = \left[1 - y_{ik} f_k(\mathbf{x}_{ik})\right]_+,$$

where $f_k(\mathbf{x}_{ik}) = (\mathbf{w}_0 + \mathbf{v}_k)\cdot\mathbf{x}_{ik} + b_0$. We call them CPRA-LR and CPRA-SVM respectively.

In this problem, λ_1 and λ_2 are regularization parameters. By adjusting their values, we control the degree of parameter sharing among different relations. The larger the ratio $\frac{\lambda_1}{\lambda_2}$ is, the more we believe that all $\mathbf{w}_t$ should conform to the common model $\mathbf{w}_0$, and the smaller the relation-specific weight $\mathbf{v}_t$ will be.

The multi-task learning problem can be directly linked to a standard single-task learning one, built on all training data from different relations.

Proposition 1 *Suppose the training data associated with the k-th relation, for every $k = 1, \cdots, K$, is transformed into:*

$$\widetilde{\mathbf{x}}_{ik} = [\frac{\mathbf{x}_{ik}}{\sqrt{\rho K}}, \underbrace{\mathbf{0},\cdots,\mathbf{0}}_{k-1}, \mathbf{x}_{ik}, \underbrace{\mathbf{0},\cdots,\mathbf{0}}_{K-k}],$$

where $\mathbf{0} \in \mathbb{R}^d$ is a vector whose coordinates are all zero, and $\rho = \frac{\lambda_2}{\lambda_1}$ a coupling coefficient. Consider a linear classifier for the transformed data $\widetilde{f}(\widetilde{\mathbf{x}}) = \widetilde{\mathbf{w}}\cdot\widetilde{\mathbf{x}} + \widetilde{b}$, with $\widetilde{\mathbf{w}}$ and $\widetilde{b}$ constructed as:

$$\widetilde{\mathbf{w}} = [\sqrt{\rho K}\mathbf{w}_0, \mathbf{v}_1, \cdots, \mathbf{v}_K] \;\; \text{and} \;\; \widetilde{b} = b_0.$$

Then the objective function of Problem 1 is equivalent to:

$$\mathcal{L} = \sum_{k=1}^{K}\sum_{i=1}^{N_k} \widetilde{\ell}\left(\widetilde{\mathbf{x}}_{ik}, y_{ik}\right) + \widetilde{\lambda}\|\widetilde{\mathbf{w}}\|_2^2,$$

where $\widetilde{\ell} = \log(1 + \exp(-y_{ik}\widetilde{f}(\widetilde{\mathbf{x}}_{ik})))$ is a logistic loss for CPRA-LR, and $\widetilde{\ell} = [1 - y_{ik}\widetilde{f}(\widetilde{\mathbf{x}}_{ik})]_+$ a hinge loss for CPRA-SVM; and $\widetilde{\lambda} = \frac{\lambda_1}{K}$.

That means, after transforming data from different relations into a unified representation, Problem 1 is equivalent to a standard single-task learning problem, built on the transformed data from all the relations. So it can easily be solved by existing tools such as LR or SVM.

5 Experiments

In this section we present empirical evaluation of CPRA in the KB completion task.

5.1 Experimental Setups

We create our data on the basis of FB15K (Bordes et al., 2011)[4], a relatively dense subgraph of Freebase containing 1,345 relations and the corresponding triples.

KB graph construction. We notice that in most cases FB15K encodes a relation and its reverse relation at the same time. That is, once a new fact is observed, FB15K creates two triples for it, *e.g.*, (x, `film/edited-by`, y) and (y, `editor/film`, x). Reverse relations provide no additional knowledge. They may even hurt the performance of PRA-style methods. Actually, to enhance graph connectivity, PRA-style methods usually automatically add an inverse version for each relation in a KB (Lao and Cohen, 2010; Lao et al., 2011). That is, for each observed triple (h, r, t), another triple (t, r^{-1}, h) is constructed and added to the KB. Consider the prediction of a relation, say `film/edited-by`. In the training phase, we could probably find that every two entities connected by this relation are also connected by the path `editor/film`$^{-1}$, and hence assign an extremely high weight to it.[5] However, in the testing phase, for any entity pair (x, y) such that (y, `editor/film`, x) has not been encoded, we might not even find that path and hence could always make a negative prediction.[6]

For this reason, we remove reverse relations in FB15K. Specifically, we regard r_2 to be a reverse relation of r_1 if the triple (t, r_2, h) holds whenever (h, r_1, t) is observed, and we randomly discard

[3] It implicitly assumes that all the relations have the same proportion of positive instances. This assumption actually holds since given any relation we can always generate the same number of negative instances for each positive one. We set this number to 4 in our experiments.

[4] https://everest.hds.utc.fr/doku.php?id=en:smemlj12

[5] For every observed triple (x, `film/edited-by`, y), FB15K also encodes (y, `editor/film`, x), for which (x, `editor/film`$^{-1}$, y) is further constructed.

[6] Note that such test cases are generally more meaningful: if we already know (y, `editor/film`, x), predicting (x, `film/edited-by`, y) could be trivial.

one of the two relations.[7] As such, we keep 774 out of 1,345 relations in FB15K, covering 14,951 entities and 327,783 triples. Then we build a graph based on this data and use it as input to CPRA (and our baseline methods).

Labeled instance generation. We select 171 relations to test our methods. To do so, we pick 10 popular domains, including award, education, film, government, location, music, olympics, organization, people, and tv. Relations in these domains with at least 50 triples observed for them are selected. For each of the 171 relations, we split the associated triples into roughly 80% training, 10% validation, and 10% testing. Since the triple number varies significantly among the relations, we allow at most 200 validation/testing triples for each relation, so as to make the test cases as balanced as possible. Note that validation and testing triples are not used for constructing the graph.

We generate positive instances for each relation directly from these triples. Given a relation r and a triple (h, r, t) observed for it (training, validation, or testing), we take the pair of entities (h, t) as a positive instance for that relation. Then we follow (Shi and Weninger, 2015; Krompaß et al., 2015) to generate negative instances. Given each positive instance (h, t) we generate four negative ones, two by randomly corrupting the head h, and the other two the tail t. To make the negative instances as difficult as possible, we corrupt a position using only entities that have appeared in that position. That means, given the relation `capitalOf` and the positive instance (`Paris`, `France`), we could generate a negative instance (`Paris`, `UK`) but never (`Paris`, `NBA`), since `NBA` never appears as a tail entity of the relation. We further ensure that the negative instances do not overlap with the positive ones.

Feature extraction and computation. Given the labeled instances, we extract path features for them using the code provided by Shi and Weninger (2015)[8]. It is a depth-first search strategy that enumerates all paths between two entities. We set the maximum path length to be $\ell = 3$. There are about 8.2% of the labeled instances for which no path could be extracted. We remove such cases, giving on average about 5,250 training, 323 validation, and 331 testing instances per relation. Then we remove paths that appear only once in each relation, getting 5,515 features on average per relation. We simply compute the value of each feature as its frequency in an instance. Table 1 lists the statistics of the data used in our experiments.

# Relations	774
# Entities	14,951
# Triples	327,783
# Relations tested	171
# Avg. training instances/relation	5,250
# Avg. validation instances/relation	323
# Avg. testing instances/relation	331
# Avg. features/relation	5,515

Table 1: Statistics of the data.

Evaluation metrics. As evaluation metrics, we use mean average precision (MAP) and mean reciprocal rank (MRR), following recent work evaluating KB completion performance (West et al., 2014; Gardner and Mitchell, 2015). Both metrics evaluate some ranking process: if a method ranks the positive instances before the negative ones for each relation, it will get a high MAP or MRR.

Baseline methods. We compare CPRA to traditional single-task PRA. CPRA first groups the 171 relations into clusters, and then learns classifiers jointly for relations within the same cluster. We implement two versions of it: CPRA-LR and CPRA-SVM. As we have shown in Proposition 1, both of them could be solved by standard classification tools. PRA learns an individual classifier for each of the relations, using LR or SVM classification techniques, denoted by PRA-LR or PRA-SVM. We use LIBLINEAR (Fan et al., 2008)[9] to solve the LR and SVM classification problems. For all these methods, we tune the cost c in the range of $\{2^{-5}, 2^{-4}, \cdots, 2^{4}, 2^{5}\}$. And we set the coupling coefficient $\rho = \frac{\lambda_2}{\lambda_1}$ in CPRA in the range of $\{0.1, 0.2, 0.5, 1, 2, 5, 10\}$.

We further compare CPRA to TransE, a widely adopted embedding-based method (Bordes et al., 2013). TransE learns vector representations for entities and relations (*i.e.*, embeddings), and uses the learned embeddings to determine the plausibility of missing facts. Such plausibility can then be used to rank the labeled instances. We implement TransE using the code provided by Bordes et al. (2013)[10]. To learn embeddings, we take as input the triples used to construct the graph (from which CPRA and PRA extract their paths). We tune the embedding dimension in $\{20, 50, 100\}$, the margin in $\{0.1, 0.2, 0.5, 1, 2, 5\}$, and the learning rate

[7] We still add an inverse version for the relation kept during path extraction.

[8] https://github.com/nddsg/KGMiner

[9] http://www.csie.ntu.edu.tw/ cjlin/liblinear

[10] https://github.com/glorotxa/SME

film/casting-director film/cinematography film/costume-design-by film/art-direction-by film/crewmember film/set-decoration-by film/production-design-by film/edited-by film/written-by film/story-by	gov-jurisdiction/dist-represent location/contain location/adjoin us-county/county-seat county-place/county location/partially-contain region/place-export
org/place-founded org/headquarter-city org/headquarter-state org/geographic-scope org/headquarter-country org/service-location	country/divisions country/capital country/fst-level-divisions country/snd-level-divisions admin-division/capital
tv/tv-producer tv/recurring-writer tv/program-creator tv/regular-appear-person tv/tv-actor	music-group-member/instrument music-artist/recording-role music-artist/track-role music-group-member/role

Table 2: Six largest clusters of relations (with the stopping criterion $\delta = 0.5$).

in $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1\}$. For details please refer to (Bordes et al., 2013). For each of these methods, we select the optimal configuration that leads to the highest MAP on the validation set and report its performance on the test set.

5.2 Relation Clustering Results

We first test the effectiveness of our agglomerative strategy (Section 4.2) in relation clustering. With the stopping criterion $\delta = 0.5$, 96 out of the 171 relations are grouped into clusters which contain at least two relations. Each of these 96 relations will later be learned jointly with some other relations. The other 75 relations cannot be merged, and will still be learned individually. Table 2 shows the six largest clusters discovered by our algorithm. Relations in each cluster are arranged in the order they were merged. The results indicate that our algorithm can effectively identify coherent clusters in which relations are highly correlated to each other. For example, the top left cluster describes relations between a film and its crew members, and the middle left between an organization and a location.

During clustering we might obtain clusters that contain too many relations and hence too many training instances for our CPRA model to learn efficiently. We split such clusters into sub-clusters, either according to the domain (*e.g.*, the film cluster and tv cluster) or randomly (*e.g.*, the two location clusters on the top right).

5.3 KB Completion Results

We further test the effectiveness of our multi-task learning strategy (Section 4.3) in KB completion. Table 3 gives the results on the 96 relations that are actually involved in multi-tasking learning (*i.e.*, grouped into clusters with size larger than one).[11] The 96 relations are grouped into 29 clusters, and relations within the same cluster are learned jointly. Table 3 reports (i) MAP and MRR within each cluster and (ii) overall MAP and MRR on the 96 relations. Numbers marked in bold type indicate that CPRA-LR/SVM outperforms PRA-LR/SVM, within a cluster (with its ID listed in the first column) or on all the 96 relations (ALL). We judge statistical significance of the overall improvements achieved by CPRA-LR/SVM over PRA-LR/SVM and TransE, using a paired t-test. The average precision (or reciprocal rank) on each relation is used as paired data. The symbol "$**$" indicates a significance level of $p < 0.0001$, and "$*$" a significance level of $p < 0.05$.

From the results, we can see that (i) CPRA outperforms PRA (using either LR or SVM) and TransE on the 96 relations (ALL) in both metrics. All the improvements are statistically significant, with a significance level of $p < 0.0001$ for MAP and a significance level of $p < 0.05$ for MRR. (ii) CPRA-LR/SVM outperforms PRA-LR/SVM in 22/24 out of the 29 clusters in terms of MAP. Most of the improvements are quite substantial. (iii) Improving PRA-LR and PRA-SVM in terms of MRR could be hard, since they already get the best performance (MRR = 1) in 19 out of the 29 clusters. But even so, CPRA-LR/SVM still improves 7/8 out of the remaining 10 clusters. (iv) The PRA-style methods perform substantially better than the embedding-based TransE model in most of the 29 clusters and on all the 96 relations. This observation demonstrates the superiority of observed features (*i.e.*, PRA paths) over latent features.

Table 4 further shows the top 5 most discriminative paths (*i.e.*, features with the highest weights) discovered by PRA-SVM (left) and CPRA-SVM (right) for each relation in the 6th cluster.[12] The average precision on each relation is also provid-

[11] The other 75 relations are still learned individually. So CPRA and PRA perform the same on these relations. The MAP values on these 75 relations are 0.6360, 0.6558, 0.6543 for TransE, PRA-LR, and PRA-SVM respectively, and the MRR values are 0.9049, 0.9033, and 0.9013 respectively.

[12] This is one of the largest clusters on which CPRA-SVM improves PRA-SVM substantially.

	MAP					MRR				
	TransE	PRA-LR	CPRA-LR	PRA-SVM	CPRA-SVM	TransE	PRA-LR	CPRA-LR	PRA-SVM	CPRA-SVM
1	0.5419	0.5160	**0.5408**	0.4687	**0.5204**	0.7500	0.8333	**1.0000**	0.7778	**0.8333**
2	0.7480	0.7888	0.7807	0.8010	**0.8092**	1.0000	1.0000	1.0000	1.0000	1.0000
3	0.4624	0.4625	**0.4788**	0.4634	0.4560	0.8333	1.0000	1.0000	1.0000	0.8333
4	0.5495	0.5378	**0.5423**	0.5385	**0.5460**	0.7667	0.6400	**0.7000**	0.7167	0.7000
5	0.5164	0.5789	**0.6030**	0.5891	**0.6072**	0.8333	0.6667	**1.0000**	0.8333	**1.0000**
6	0.6918	0.7733	**0.7950**	0.7369	**0.8084**	1.0000	0.8333	0.8333	0.8056	**0.9167**
7	0.7381	0.7531	**0.7754**	0.7456	0.7414	0.7500	1.0000	1.0000	1.0000	1.0000
8	0.4258	0.5180	**0.5446**	0.3162	**0.4606**	1.0000	1.0000	1.0000	0.3056	**0.7500**
9	0.6353	0.7879	0.7708	0.7680	**0.7685**	0.7500	1.0000	1.0000	1.0000	1.0000
10	0.8615	0.7773	0.7738	0.7618	0.7507	1.0000	1.0000	1.0000	1.0000	1.0000
11	0.4549	0.5814	**0.6014**	0.5717	**0.5896**	0.8333	1.0000	1.0000	0.8750	**1.0000**
12	0.6202	0.7187	**0.7479**	0.7455	**0.7457**	0.7500	0.5833	**1.0000**	1.0000	1.0000
13	0.5530	0.6681	**0.6716**	0.6373	**0.6502**	0.6667	1.0000	1.0000	1.0000	1.0000
14	0.5082	0.4360	**0.5280**	0.4715	**0.5806**	0.3750	0.6667	0.6250	1.0000	1.0000
15	0.9881	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
16	0.5324	0.6818	**0.6863**	0.6522	**0.6705**	0.8750	1.0000	1.0000	1.0000	1.0000
17	0.3759	0.3351	**0.3593**	0.3273	0.3219	0.6111	0.5667	**0.7778**	0.5111	**0.6667**
18	0.9423	0.9968	**1.0000**	0.9947	**0.9975**	1.0000	1.0000	1.0000	1.0000	1.0000
19	0.7903	0.8376	0.8310	0.8296	**0.8328**	0.8714	0.9286	0.8571	0.8571	0.8571
20	0.7920	0.8285	**0.8746**	0.8491	**0.8754**	1.0000	1.0000	1.0000	1.0000	1.0000
21	0.4885	0.5869	0.5799	0.5554	**0.5952**	0.6250	1.0000	1.0000	1.0000	1.0000
22	0.7894	0.8371	**0.8486**	0.8371	**0.8374**	1.0000	1.0000	1.0000	1.0000	1.0000
23	0.7123	0.7848	**0.8191**	0.7811	**0.7957**	0.9500	1.0000	1.0000	1.0000	1.0000
24	0.5982	0.7923	**0.8048**	0.8204	**0.8220**	1.0000	1.0000	1.0000	1.0000	1.0000
25	0.6223	0.8723	0.8723	0.7785	**0.8109**	0.7500	1.0000	1.0000	1.0000	1.0000
26	0.5253	0.5377	**0.5685**	0.5337	**0.5447**	0.8750	0.8125	**0.8750**	0.7083	**0.8333**
27	0.8763	0.6890	**0.8124**	0.7014	**0.8016**	1.0000	0.6667	**1.0000**	0.7500	**1.0000**
28	0.7588	0.8131	**0.8154**	0.8130	**0.8146**	1.0000	1.0000	1.0000	1.0000	1.0000
29	0.4894	0.5921	**0.6543**	0.6093	**0.6566**	0.7500	1.0000	1.0000	1.0000	1.0000
ALL	0.6540	0.7058	**0.7254****	0.6943	**0.7162****	0.8682	0.9061	**0.9436***	0.8982	**0.9358***

Table 3: KB completion results on the 96 relations that have been grouped into clusters with size larger than one (with the stopping criterion $\delta = 0.5$), and hence involved in multi-tasking learning.

ed. We can observe that (i) CPRA generally discovers more predictive paths than PRA. Almost all the top paths discovered by CPRA are easily interpretable and provide sensible reasons for the final prediction, while some of the top paths discovered by PRA are hard to interpret and less predictive. Take `org/place-founded` as an example. All the 5 CPRA paths are useful to predict the place where an organization was founded, *e.g.*, the 3rd one tells that "the organization headquarter in a city which is located in that place". However, the PRA path "common/class → common/class^{-1} → film/debut-venue" is hard to interpret and less predictive. (ii) For the 1st/4th/6th relation on which PRA gets a low average precision, CPRA learns almost completely different top paths and gets a substantially higher average precision. While for the other relations (2nd/3rd/5th) on which PRA already performs well enough, CPRA learns similar top paths and gets a comparable average precision. We have conducted the same analyses with CPRA-LR and PRA-LR, and observed similar phenomena. All these observations demonstrate the superiority of CPRA, in terms of not only predictive accuracy but also model interpretability.

6 Conclusion

In this paper we have studied the path ranking algorithm (PRA) from the viewpoint of multi-task learning. We have designed a novel multi-task learning framework for PRA, called coupled PRA (CPRA). The key idea of CPRA is to (i) automatically discover relations highly correlated to each other through agglomerative clustering, and (ii) effectively couple the prediction of such relations through multi-task learning. By coupling different relations, CPRA takes into account relation associations and enables implicit data sharing among them. We have tested CPRA on benchmark data created from Freebase. Experimental results show that CPRA can effectively identify coherent clusters in which relations are highly correlated. By further coupling such relations, CPRA significantly outperforms PRA, in terms of both predictive

org/place-founded (0.4920 vs. 0.6750)	
org/headquarter-city common/class → common/class^{-1} → film/debut-venue common/class → common/class^{-1} → sports-team/location employer/job-title → employer/job-title^{-1} → location/contain music-artist/label^{-1} → person/place-of-birth	location/contain^{-1} org/headquarter-city org/headquarter-city → location/contain^{-1} org/headquarter-state → location/contain org/headquarter-city → bibs-location/state
org/headquarter-city (0.9014 vs. 0.9141)	
location/contain^{-1} org/place-founded org/headquarter-state → location/contain org/child^{-1} → org/child → org/place-founded sports-team/location	location/contain^{-1} org/headquarter-state → location/contain org/place-founded org/child^{-1} → org/child → org/place-founded industry/company^{-1} → industry/company → org/place-founded
org/headquarter-state (0.9522 vs. 0.9558)	
location/contain^{-1} org/headquarter-city → location/contain^{-1} org/headquarter-city → bibs-location/state org/headquarter-city → county-place/county → location/contain^{-1} org/headquarter-city → location/contain^{-1} → location/contain^{-1}	location/contain^{-1} org/headquarter-city → location/contain^{-1} org/headquarter-city → bibs-location/state org/headquarter-city org/place-founded
org/geographic-scope (0.5252 vs. 0.6075)	
common/class → common/class^{-1} → location/vacationer^{-1} common/class → common/class^{-1} → country/languages^{-1} common/class → common/class^{-1} → gov-jurisdiction/gov-body^{-1} common/class → common/class^{-1} → region/currency-of-gdp^{-1} politician/party^{-1} → person/nationality → location/adjoins	location/contain^{-1} org/headquarter-city → location/contain^{-1} location/contain^{-1} → location/contain^{-1} org/place-founded → location/contain^{-1} org/headquarter-city → location/contain^{-1} → location/contain^{-1}
org/headquarter-country (0.9859 vs. 0.9938)	
org/headquarter-city → airline/city-served^{-1} → org/service-location org/headquarter-city → admin-area/child^{-1} → region/place-export org/headquarter-city → country/divisions^{-1} → region/place-export org/headquarter-city → film/feat-location^{-1} → film/feat-location org/headquarter-city → gov-jurisdiction/title → employer/job-title^{-1}	location/contain^{-1} org/headquarter-city → location/contain^{-1} org/headquarter-city → county-place/county → location/contain^{-1} location/contain^{-1} → location/contain^{-1} org/place-founded → location/contain^{-1}
org/service-location (0.5644 vs. 0.7044)	
org/headquarter-city → country/divisions^{-1} org-extra/service-location film/production-company^{-1} → film/subjects → admin-area/child^{-1} org/legal-structure → entry/taxonomy → entry/taxonomy^{-1} airline/city-served → region/currency → region/currency-of-gdp^{-1}	org/headquarter-city → location/contain^{-1} org/headquarter-city → county-place/county → location/contain^{-1} location/contain^{-1} → location/contain^{-1} org/place-founded → location/contain^{-1} org-extra/service-location

Table 4: Top paths given by PRA-SVM (left) and CPRA-SVM (right) for each relation in the 6th cluster.

accuracy and model interpretability.

This is the first work that investigates the possibility of multi-task learning with PRA, and we just provide a very simple solution. There are still many interesting topics to study. For instance, the agglomerative clustering strategy can only identify highly correlated relations, *i.e.*, those sharing a lot of common paths. Relations that are only loosely correlated, *e.g.*, those sharing no common paths but a lot of sub-paths, will not be identified. We would like to design new mechanisms to discover loosely correlated relations, and investigate whether coupling such relations still provides benefits. Another example is that the current method is a two-step approach, performing relation clustering first and then relation coupling. It will be interesting to study whether one can merge the clustering step and the coupling step so as to have a richer inter-task dependent structure. We will investigate such topics in our future work.

Acknowledgments

We would like to thank Baoxu Shi for providing the code for path extraction. We would also like to thank the anonymous reviewers for their valuable comments and suggestions. This work is supported by the National Natural Science Foundation of China (grant No. 61402465), the Strategic Priority Research Program of the Chinese Academy of Sciences (grant No. XDA06030200), and the Microsoft Research Asia StarTrack Program. This work was done when Quan Wang was a visiting researcher at Microsoft Research Asia.

References

Rie Kubota Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Reseach*, 6:1817–1853.

Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. 2007. Multi-task feature learning. In *Advances in Neural Information Processing Systems*, pages 41–48.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, pages 1247–1250.

Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. 2011. Learning structured embeddings of knowledge bases. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence*, pages 301–306.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*, pages 2787–2795.

Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka Jr, and Tom M. Mitchell. 2010. Toward an architecture for never-ending language learning. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, pages 1306–1313.

Rich Caruana. 1997. Multitask learning. *Machine Learning*, 28(1):41–75.

Olivier Chapelle, Pannagadatta Shivaswamy, Srinivas Vadrevu, Kilian Weinberger, Ya Zhang, and Belle Tseng. 2010. Multi-task learning for boosting with application to web search ranking. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1189–1198.

Eunsol Choi, Tom Kwiatkowski, and Luke Zettlemoyer. 2015. Scalable semantic parsing with partial ontologies. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 1311–1320.

Silviu Cucerzan. 2007. Large-scale named entity disambiguation based on Wikipedia data. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 708–716.

Lei Cui, Xilun Chen, Dongdong Zhang, Shujie Liu, Mu Li, and Ming Zhou. 2013. Multi-domain adaptation for SMT using multi-task learning. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1055–1065.

Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. 2014. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 601–610.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 1723–1732.

Theodoros Evgeniou and Massimiliano Pontil. 2004. Regularized multi-task learning. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 109–117.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. Liblinear: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.

Matt Gardner and Tom Mitchell. 2015. Efficient and expressive knowledge base completion using subgraph feature extraction. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1488–1498.

Matt Gardner, Partha Pratim Talukdar, Bryan Kisiel, and Tom Mitchell. 2013. Improving learning and inference in a large knowledge-base using latent syntactic cues. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 833–838.

Matt Gardner, Partha Talukdar, Jayant Krishnamurthy, and Tom Mitchell. 2014. Incorporating vector space similarity in random walk inference over knowledge bases. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 397–406.

Shu Guo, Quan Wang, Lihong Wang, Bin Wang, and Li Guo. 2015. Semantically smooth knowledge graph embedding. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 84–94.

Jingrui He, Yan Liu, and Qiang Yang. 2014. Linking heterogeneous input spaces with pivots for multi-task learning. In *Proceedings of the 2014 SIAM International Conference on Data Mining*, pages 181–189.

Shangpu Jiang, Daniel Lowd, and Dejing Dou. 2012. Learning to refine an automatically extracted knowledge base using markov logic. In *Proceedings of the 2012 IEEE International Conference on Data Mining*, pages 912–917.

Jing Jiang. 2009. Multi-task transfer learning for weakly-supervised relation extraction. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1012–1020.

Denis Krompaß, Stephan Baier, and Volker Tresp. 2015. Type-constrained representation learning in knowledge graphs. In *Proceedings of the 13th International Semantic Web Conference*, pages 640–655.

Ni Lao and William W. Cohen. 2010. Relational retrieval using a combination of path-constrained random walks. *Machine Learning*, 81(1):53–67.

Ni Lao, Tom Mitchell, and William W. Cohen. 2011. Random walk inference and learning in a large scale knowledge base. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 529–539.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, et al. 2014. Dbpedia: A large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web Journal*.

Arvind Neelakantan, Benjamin Roth, and Andrew M-cCallum. 2015. Compositional vector space models for knowledge base completion. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 156–166.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th International Conference on Machine Learning*, pages 809–816.

Maximilian Nickel, Xueyan Jiang, and Volker Tresp. 2014. Reducing the rank in relational factorization models by including observable patterns. In *Advances in Neural Information Processing Systems*, pages 1179–1187.

Jay Pujara, Hui Miao, Lise Getoor, and William Cohen. 2013. Knowledge graph identification. In *Proceedings of the 11th International Semantic Web Conference*, pages 542–557.

Matthew Richardson and Pedro Domingos. 2006. Markov logic networks. *Machine Learning*, 62(1-2):107–136.

Michael Schuhmacher and Simone Paolo Ponzetto. 2014. Knowledge-based graph document modeling. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, pages 543–552.

Rico Sennrich, Holger Schwenk, and Walid Aransa. 2013. A multi-domain translation model framework for statistical machine translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 832–840.

Baoxu Shi and Tim Weninger. 2015. Fact checking in large knowledge graphs: A discriminative predict path mining approach. In *arXiv:1510.05911*.

Kristina Toutanova and Danqi Chen. 2015. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and Their Compositionality*.

Robert West, Evgeniy Gabrilovich, Kevin Murphy, Shaohua Sun, Rahul Gupta, and Dekang Lin. 2014. Knowledge base completion via search-based question answering. In *Proceedings of the 23rd International Conference on World Wide Web*, pages 515–526.